A Divisive Initialisation Method for Clustering Algorithms

Clara Pizzuti, Domenico Talia and Giorgio Vonella

ISI-CNR
c/o DEIS, Univ. della Calabria
87036 Rende (CS), Italy
{pizzuti,talia}@si.deis.unical.it

Abstract. A method for the initialisation step of clustering algorithms is presented. It is based on the concept of cluster as a high density region of points. The search space is modelled as a set of d-dimensional cells. A sample of points is chosen and located into the appropriate cells. Cells are iteratively split as the number of points they receive increases. The regions of the search space having a higher density of points are considered good candidates to contain the true centers of the clusters. Preliminary experimental results show the good quality of the estimated centroids with respect to the random choice of points. The accuracy of the clusters obtained by running the K-Means algorithm with the two different initialisation techniques - random starting centers chosen uniformly on the datasets and centers found by our method - is evaluated and the better outcome of the K-Means by using our initialisation method is shown.

1 Introduction

Clustering is a data analysis unsupervised technique which searches to separate data items (objects), having similar characteristics, in constituent groups. The importance of this technique has been recognized in diverse fields such as sociology, biology, statistics, artificial intelligence and information retrieval. In the last few years clustering has been identified as one of the main data mining tasks [7]. Many of the proposed definitions of what constitutes a cluster are based on the concepts of *similarity* and *distance* among the objects. A more general approach consists in considering a data item as a point in a d-dimensional space, with each of the d variables being represented by one of the axes of this space. The variable values of each data item thus define a d-dimensional coordinate in this space. Clusters can be described as continuous regions of this space with a relatively high density of points, separated from other clusters by regions with a relatively low density of points. Clusters described in this way directly reflect the process we use to detect clusters visually in two or three dimensions and lead to see clustering as a density estimation problem. Density estimation has been addressed in statistics and has been recognized as a challenging problem [14].

Several approaches have been defined for data clustering. Among them the *partitioning methods* produce a partition of a set of objects into K clusters, where the number K of groups is given a priori. The performance of these methods is strongly influenced by the choice of the initial clusters. The nearer are the initial starting points to the true centers of the clusters, the better is the quality of the clustering and the convergence of the method is accelerated.

In this paper a divisive method for the initialisation step of clustering algorithms is presented. The method provides an estimation of the centroids of clusters. It is based on the concept of cluster as a high density region of points and on the intuition that subsampling the data will naturally bias the sample to representatives near the more dense regions [3, 8]. The search space is modelled as a set of d-dimensional cells. A sample of points is chosen and located into the appropriate cells. Cells are iteratively split as the number of points assigned to them increases. In such a way the attention is focused on those regions of the search space having a higher density of points and, thus, candidates to contain the true centers of the clusters. Preliminary experimental results show the good quality of the estimated centroids with respect to random choice of points. Furthermore, the accuracy of the clusters obtained by running the K-Means algorithm with the two different initialisations - random starting centers chosen uniformly on the datasets and centers found by our method - is evaluated. The results point out the better performances of K-Means when estimated centroids are provided by our method.

The paper is organized as follows: section 2 gives a brief survey of clustering methods; section 3 presents our divisive method to estimate centroids; section 4 shows the effectiveness of the proposed approach on a set of synthetic data.

2 Background and related work

Several algorithms for clustering data have been proposed [1, 5, 11, 13, 15]. In [10] an extensive overview of clustering algorithms is presented. One of the approaches to clustering is the *partitional* approach. A partitional clustering algorithm produces a partition of a set of objects into K clusters by optimizing a criterion function. Most of these approaches assume that the number K of groups has been given a priori. They can be formulated as three main steps: 1) initialisation of clusters, 2)allocation of the objects to clusters, 3) reallocation of objects to other clusters after the initial grouping process has been completed. The differences between the methods mainly lie in (1) and (3). The techniques for initialisation of clusters find K points in the d-dimensional space and use them as initial estimates of cluster centers. The choice of these points is an important aspect of this approach since it strongly biases the performances of the methods [10]. Among the partitioning methods, the *direct K-Means clustering* [12, 1] is a well-known and effective method for many practical applications. In the direct K-Means clustering the n data points must be partitioned into $C_1, \ldots, C_k$ clusters, where each cluster C_j contains n_j data points and each data point belongs to only one cluster. The criterion function adopted in this tecnhique is the squared error criterion. The *centroid* of the cluster C_j is defined as the mean of the objects it contains. The *square-error* e_j^2 of each partition C_j is the sum of the squared Euclidean distances between each data point in C_j and its centroid. The *square-error* of the partition is the sum of the square errors of all partitions $E = \sum_{j=1}^{K} e_j^2$. The smaller the error, the better is the quality of the partitioning. The direct K-Means algorithm starts by randomly choosing the initial centroids and then by assigning each data point to the cluster C_j with the nearest centroid. The new centroids and the square-error E are computed and the method stops either when E does not change significantly or no movement of points from one cluster to another occurs. A good choice of the initial centroids, sufficiently near to the actual centroids, can drastically reduce the number of steps needed by the K-Means to converge and can avoid the problem of empty clusters due to the choice of starting points far from the centroids.

In literature there are few proposals on initialisation methods [6,2,3,8]. The most recent is the one proposed by Bradley and Fayyad in [3] where they present an algorithm for refining an initial starting condition. The method chooses a given number of subsamples from the dataset and groups them by using the K-Means. Each subsample gives a solution CM_i. The set of such solutions is then clustered via the K-Means initialised with CM_i giving a solution FM_i. Finally, the FM_i having the minimal squared error is chosen as the refined initial point.

In the next section we present an initialisation method which, analogously to [3] works on a sample of data, but exploits the concept of density to focus attention on those parts of the search space that contain a high frequency count of the sampled points. These parts are iteratively divided into smaller subparts and assumed as possible candidates to contain the true centers.

3 The method

We now present a general method for providing a good estimation of the centroids of clusters for clustering algorithms. The method exploits the concept of density in the search space, by focusing the attention on those regions having a high density in the data points.

Given the dataset D of n data points X_i, in d dimensions, a cell $I_D \subset R^d$ containing all the points of D is defined in the following way.

$$I_D = [X_{min}, X_{max}] = \{X_i \in D \mid X_{min}^{(j)} \leq X_i^{(j)} \leq X_{max}^{(j)}, 1 \leq j \leq d\}$$

where $X_{min}^{(j)}$=min$\{X_i^{(j)}, 1 \leq i \leq n\}$ and $X_{max}^{(j)}$=max$\{X_i^{(j)}, 1 \leq i \leq n\}$.
The search space can thus be represented by I_D and the problem of finding K centroids can be formulated as the problem of finding K subcells of I_D having the highest density of points. The *Divisive Initialisation* algorithm, shown in Figure 1, initially partitions the cell I_D in 2^d subcells by dividing each side of I_D into two equal parts. It takes as input data points D, number J of different subsamplings to do, frequency F of times that the cells must be considered for

```
function Divisive Initialisation(D,J,F,K)
Compute I_D
Let CL be the set of 2^d cells generated by dividing I_D
 C = ∅
 repeat
  for i=1,...,J do
   let S_i a subsample of D
   C_i = findCentroids(S_i, F, CL)
   C = C ∪ C_i
  end for
 until | C |≥ K
if | C |> K do
 let C_S be K points randomly selected from C
 while (C − C_S ≠ ∅) do
   select a point P from C_S
   compute the nearest point Q to P
   P = mean(P, Q)
   if Q ∈ C_S then select another point P_1
   in C − C_S and substitute Q by P_1
    delete P_1 from C − C_S
   else delete Q from C − C_S
   end if
 end while
 return C_S
 end if
 else return C
```

```
function findCentroids(S_i, F, CL)
  maxlevel =| S_i | /F
  count=0
  set the level of splitting of each cell to zero
  for l = 1 to | S_i | do
   read a point from S_i
   assign it to the appropriate cell of CL
   if count = F do
    find the cell C_max containing the highest number of points
    divide C_max into 2^d subcells
    reallocate the points of C_max
    increment by one the level of splitting of these subcells
    count = 0
   end if
   else
    count = count + 1
  end for
  PM = ∅
  for each cell CL(h) in CL having splitting
  level equal to maxlevel or maxlevel − 1 do
   compute the mean point P_h of the points contained in CL(h)
   PM = PM ∪ P_h
  end for
return (PM)
```

Fig. 1. Divisive method and Centroids found for each subsample.

splitting and number K of clusters. It randomly chooses J subsamples of data points D, $S_i, i = 1, \dots, J$ and, for each of them, finds C_i centroids by calling the function *findCentroids*, shown in Figure 1. The sets $C_i, i = 1, \dots, J$ are then considered for possible fusion of points that could be representatives of the same cluster. The function *findCentroids* at the beginning assigns to each cell a *level of splitting* equal to zero. This level is incremented by one, every F point read, only for the subcells generated by splitting the cell that received the highest number of points. A graphical view of the iterative splitting of the dataset DS2 (described in the next section) with one subsample is shown in Figure 2. It is worth noting that regions not containing points from the subsample are completely ignored during the division process. In fact, since no point from the subsample has been located in these regions, they will never be considered for the division process. The points belonging to S_i are located in the appropriate subcell and when all the sample of points has been considered, the cells having a splitting level equal to maxlevel or maxlevel -1 are selected and the mean points of the points they contain are computed. These mean points are considered as possible representatives of clusters.

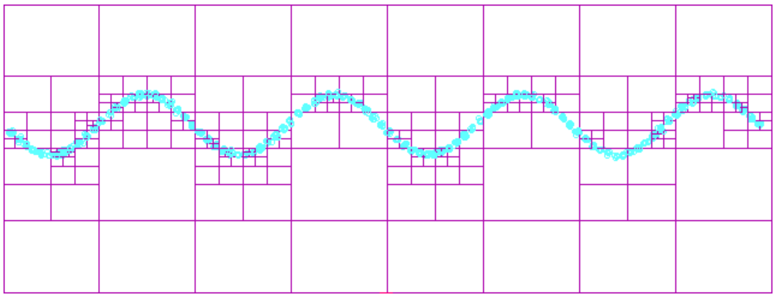


Fig. 2. Iterative splitting of the cells for DS2.

4 Experimental Results

The goal of the experiments was to evaluate the quality of the centroids estimated by our method with respect to the actual centers of two synthetic datasets, DS1 and DS2, described in [15]. Both datasets consists of K clusters of 2-dimensional data points. DS1 has the grid pattern and its centers are placed on a $\sqrt{K} \times \sqrt{K}$ grid. DS2 has the *sine* pattern and places the cluster centers on a curve of the sine function. Both DS1 and DS2 are constituted by 100000 data points grouped in 100 clusters. The size of the random subsamples we used was 1% of the full dataset and the number of subsampling was taken to be 10. The first set of experiments evaluated the closeness of the centroids estimated by the *Divisive* algorithm to the true centers of the synthetic data. Figures 3 draws the actual (identified with the symbol X), estimated (with +) and random centers (with *) for DS1. As the figure show, the centroids computed by our method are very near to the true centers. Next, we run the K-Means algorithm with two different initialisations: random starting centers chosen uniformly on the datasets and centers found by the *Divisive Initialisation* algorithm. Figure 4 visualizes the clusters obtained at the first step with random start and with estimated centers for the dataset DS1 and the same clusters after 10 steps.

Comparing these results with the actual clusters, we can observe that: (1) the location of centers by using the random initialisation is far away from the true location, whether the location of the estimated centers is very close to the actual one; (2) the number of points contained in the clusters can remarkably be different from the actual number as regard the random start; for the estimated centers, on the contrary, only few clusters move away from them. The same results have been obtained for DS2, but they have been omitted due to the lack of space. Finally, table 1 shows, for the first five iterations and then for 10th,

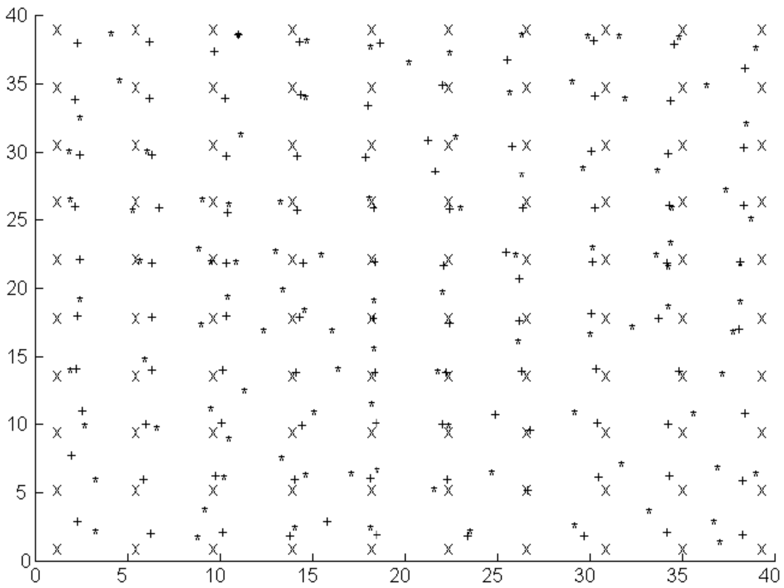
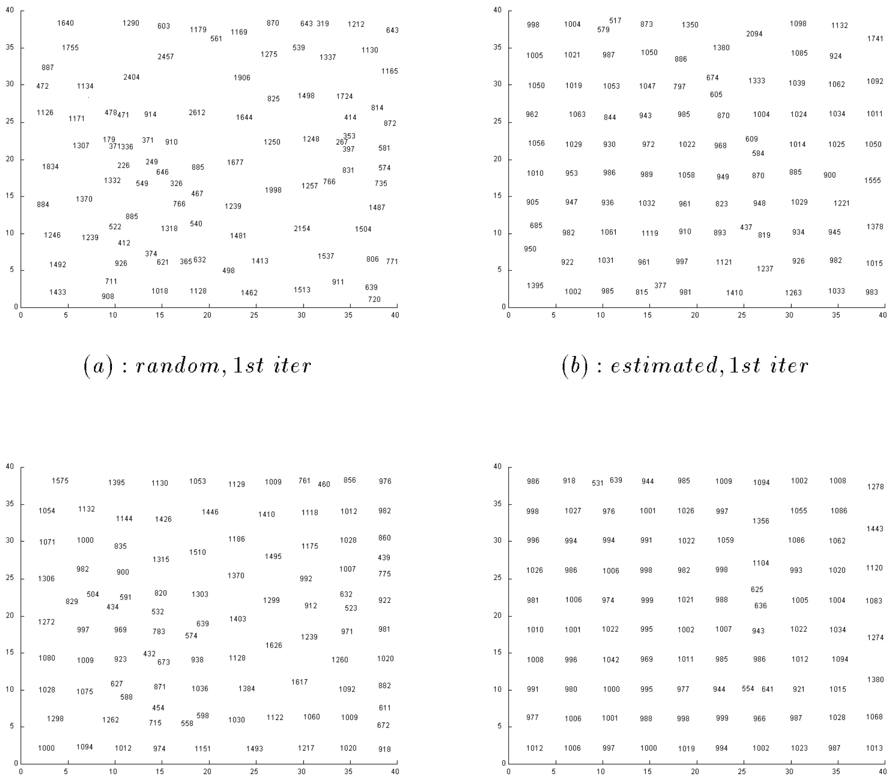


Fig. 3. Centers of DS1: X : true, + : estimated, * : random.

20th and 30th iterations, the square-error E and the difference with the square error E' of the previous step, for both the datasets DS1 and DS2, when the K-Means is initialised by our method and by a random choice of the centers. The value of the column *reduc* compares the reduction of the square-error E for the K-Means initialised with respect to a random start.

5 Conclusions and Future Work

Initialisation methods that estimate the first values of centers of clusters can play an important role in supporting clustering algorithms to accurately and efficiently identify groups of items in large data sets. Preliminary results on two synthetic datasets showed the good quality of the estimated centers. Further work, however, is still necessary to assess the effectiveness of our method. A comparison of our approach with that of Bradley and Fayyad, by using their



$(a) : random, 1st\ iter$ $(b) : estimated, 1st\ iter$

$(c) : random, 10th\ iter$ $(d) : estimated, 10th\ iter$

Fig. 4. Clusters of DS1 with random ((a) and (c)) and estimated ((b) and (d) centers for the 1st and 10th iterations.

synthetic data, need to be done and the application on real world data is to be performed.

References

1. K. Alsabti, S. Ranka, V. Singh. An Efficient K-Means Clustering Algorithm. *Proceedings of the First Workshop on High Performance Data Mining*, Orlando, Florida, 1998.
2. P.S. Bradley, O.L. Mangasarian, W.N. Street. Clustering via Concave Minimisation. *Advances in Neural Information Processing Systems 9*, Mozer, Jordan and Petsce Eds. pp. 368-374, MIT Press, 1997.
3. P.S. Bradley, U.M. Fayyad. Refining Initial Points for K-Means Clustering. *Proceedings of the Int. Conf. on Machine Learning*, pp. 91-99, Morgan Kaufmann, 1998.

Table 1. Square-errors for DS1 and DS2.

	DS1					**DS2**				
	Divisive		**random start**			**Divisive**		**random start**		
iter	**E**	**E-E'**	**E**	**E-E'**	**reduc**	**E**	**E-E'**	**E**	**E-E'**	**reduc**
1	22341823		50670020		2.26	601370		2081689		3.46
2	19255196	3086627	30930736	19739284	1.60	280649	320721	774035	1307654	2.75
3	18716145	539051	26999328	3931408	1.44	251671	28978	587834	186201	2.33
4	18518756	197389	25230965	1768363	1.36	236701	14970	539755	48079	2.28
5	18457450	61306	24211837	1019128	1.31	234800	1901	536525	3230	2.28
10	18313930	31143	22277027	322189	1.24	232709	427	447485	8352	1.92
20	17377078	10225	20228928	87325	1.16	231422	6	445371	404	1.92
30	16931927	1062	19706827	57233	1.16	231428	0	444606	3	1.92

4. R.C. Dubes, A.K. Jain. *Algorithms for Clustering Data.* Prentice Hall, 1988.
5. M. Ester, H. Kriegel, J. Sander, X. Xu. A Density Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd Int. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, 1996.
6. B. Everitt. *Cluster Analysis.* Heinemann Educational Books Ltd, London, 1977.
7. U.M. Fayyad, G. Piatesky-Shapiro, P. Smith. From Data Mining to Knowledge Discovery: an overview. In U.M. Fayyad & al. (Eds) *Advances in Knowledge Discovery and Data Mining*, pp. 1-34, AAAI/MIT Press, 1996.
8. U.M. Fayyad, C. Reina, P.S. Bradley. Initialisation of Iterative Refinement Clustering Algorithms. *Proceedings of the Int. Conf. on Knowledge Discovery and Data Mining*, AAAI Press, 1998.
9. A.K. Jain. *Algorithms for Clustering Data.* Prentice Hall, 1988.
10. A.K. Jain, M. N. Murty, P. J. Flynn. *Data Clustering : A Review.* ACM Computing Surveys, June 1999.
11. D. Judd, P. McKinley, A. Jain. Large-Scale Parallel Data Clustering. *Proceedings of the Int. Conf. on Pattern Recognition*, 1996.
12. L. Kaufman, P.J. Rousseew. *Finding Groups in Data: an Introduction to Cluster Analysis.* John Wiley & Sons, 1990.
13. R.T. Ng, J. Han. Efficient and Effective Clustering Methods for Spatial Data Mining. *Proceedings of the 20th Int. Conf. on Very Large Data Bases*, pp. 144-155, 1994.
14. B.W. Silverman. *Density Estimation for Statistics and Data Analysis.* London, Chapman and Hall, 1986.
15. T. Zhang, R. Ramakrishnan, M. Livny. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *Proceedings of the ACM SIGMOD Int. Conf. on Managment of Data*, Montreal, Canada, pp. 103-114, June 1996.